情報検索のための表記の揺れに寛容な類似尺度

山本　英子[†]　武田　善行[††]　梅村　恭司[††]

本論文では，情報検索のための表記の揺れに寛容な類似尺度を提案する．情報検索において，検索対象となるデータがさまざまな人によって記述されたものであるため，同じ事柄であっても表記が異なり，入力した文字列で意図した情報を得ることができない場合がある．人間ならば，表記が多少異なっていて（表記の揺れがあって）も柔軟に対応し，一致していると判断できるが，計算機はこの柔軟性を備えていない．表記の揺れに対応することができる尺度として編集距離が知られているが，実際にこの尺度を単純に類似尺度に変換したものを用いて情報検索を行ってみたが，性能がでなかった．そこで，本論文では，この単純な類似尺度を情報検索に適した表記の揺れに寛容な類似尺度に拡張することを試み，その結果，この拡張によって検索性能が向上したことを示す．さらに，提案する類似尺度を組み込んだ情報検索システムを構築し，多くの情報検索システムに用いられている一般的な類似尺度と同等以上の検索性能を実現できたことを示す．

キーワード:　表記の揺れ，編集距離，類似尺度，文字列重み

An IR Similarity Measure which is Tolerant for Morphological Variation

Eiko Yamamoto[†] , Yoshiyuki Takeda[††] and Kyoji Umemura[††]

In this paper, we propose a measure for information retrieval (IR). This measure is tolerant for morphological variation. When various persons describe the data to retrieve, their notations may vary even if the data describe the same topic. This variation prevents system to retrieve all of relevant documents for the input sentence. Although human can handle this variation, computers usually can not handle this. Edit distance is a well-known measure that can cope with this variation. We have used this measure for information retrieval and found that its precision is poor. Therefore, we propose to modify this similarity measure to be suitable for information retrieval. We show that this extension improves the performance. We also compared the proposed similarity measure with the popular similarity measures used in many information retrieval systems.

KeyWords: *Morphological variation, Edit distance, Similarity measure, String weight*

† 通信総合研究所, Communications Research Laboratory
†† 豊橋技術科学大学情報工学系, Dept. of Information and Computer Sciences, Toyohashi University of Technology

1　はじめに

情報検索において，検索対象となるデータはさまざまな人に記述されたものであり，同じ事柄を表す言葉であっても人によって表記が異なるために，ユーザは検索システムから意図した情報を得られないことがある．人間ならば柔軟に表記から意図を読み取り対応できるが，機械はこの柔軟性を備えていない．ここで考える表記の異なりとは，たとえば,「ウイルス」と「ウィルス」,「コンピュータ」と「コンピューター」といった一般的な表記の揺ればかりでなく，その他「機械を使って翻訳する」という事柄を表すために，ある人は「機械翻訳」，別の人は「機械による翻訳」と多少表現が異なるといった表記の違いといったあいまいな表現のことである．本研究では，このようなあいまいな表現を合わせて「表記の揺れ」と呼ぶ．情報検索においてあいまいな表現は性能低下を招く．日本語には表記の揺れが多く存在するために，日本語における情報検索は難しいものである．そこで，表記の揺れに対応できる類似尺度が必要である．これまでに，表記の揺れに対応できる尺度として，編集距離 (Korfhage 1997) が知られている．編集距離は一方の文字列をもう一方の文字列に一致させるために必要な最小限の編集操作の数である．編集操作には挿入，削除，置換があり，編集操作の数を距離として考える．このため，編集距離は二つの文字列の不一致な文字を計数する相違尺度とみることができる．そこで，本論文ではまず，この編集距離を一致する文字を計数する類似尺度に変換し，情報検索テストコレクション NTCIR1(Kando, Koyama, Oyama, Kageura, Yoshioka, Nozue, Matsumura, Kuriyama 1998; Kageura, Koyama, Yoshioka, Takasu, Nozue, Tsuji 1997) を用いて実験を行ったが，その結果は満足できるものではなかった．その原因の一つは，文字をすべて同等に扱い，文章の意味に大きく関わるような文字と表記の揺れとなりうる文字を区別せずに計数したことにあると考えた．たとえば，ひらがなは助詞や助動詞を表現するために用いられることが多く，漢字は名詞や動詞の表記に用いられるため，ひらがなの一致と漢字の一致では直感的にも重要さが異なるにもかかわらず，同じように一致した文字を計数してしまうことである．もう一つの原因は，編集距離の定義に使われている編集操作が一文字に限られていたことにあると考えた．たとえば，連続した三文字が一致した場合と不連続な三文字が一致した場合では直感的にも重要さが異なるにもかかわらず，同じように一致した文字を計数してしまうことである．

本論文では，この二つの原因を解消するために，一致した文字に対して重み付けを行い，次に一致した文字列に対応できるように，編集距離を変換した類似尺度の拡張を試みる．そして，編集距離から最終的に本論文で提案する類似尺度に到達する過程で定義する類似尺度を組み込んだシステムを構築し，類似尺度を拡張することによって表記の揺れに寛容な性質を損なうことなく，情報検索性能が向上するかを検証する．さらに，一致した文字列に対する重みをその文字列が持つ IDF に基づくスコアとするという条件の下で，類似尺度の違いによる情報検索性能の差を検証する．すなわち，本論文で提案する表記の揺れに寛容な類似尺度を組み込んだ情報検索システムと，形態素解析によって得られた単語を一致する文字列の単位とし，その単

語が持つ $tf \cdot IDF$ を重みとして累計するシステム，$ngram$ を一致する文字列の単位とし，その $ngram$ が持つ IDF を重みとして累計するシステムと比較する．実験結果において，本論文で提案する類似尺度を用いたシステムが，従来法である単語に基づくシステムや $ngram$ に基づくシステムと同等以上の検索性能を実現できたことを示す．

この論文の構成は次のとおりである．2節では，編集距離から本論文で提案する類似尺度に到達するまでの過程をその過程で定義される類似尺度とともに示す．3節では，本論文で用いる重みを明示する．4節では，本論文で行った情報検索性能を測るための実験の概要を示す．5節では，2節で定義した類似尺度の検索性能が実際に定義した順に向上しているかと表記の揺れに寛容な性質が損なわれていないかを検証する．6節では，5節の結果を踏まえ，本論文で提案する類似尺度の検索性能を測るために，比較対象としたシステムについて説明した後，検索性能の比較を行う．7節でこれまで示した実験結果から考察を述べ，最後にまとめる．

2　編集距離の類似尺度への変換とその拡張

本論文では，検索性能の低下を招く表記の揺れに寛容な類似尺度を提案する．そのために，まず編集距離を類似尺度に変換し，次に一致した文字の重みを加算する類似尺度に拡張し，最後に一致した文字列の重みを加算する類似尺度に拡張する．本節では，この三つのステップ毎に定義した類似尺度を示すことで，本論文で提案する類似尺度が考慮する性質を明示する．

2.1　編集距離の類似尺度への変換

情報検索において，検索対象となるデータに存在する表記の揺れは検索性能の低下を招くものである．そこで，本論文では，表記の揺れに対応することができる尺度としてよく知られている編集距離に基づく類似尺度を考えた．

編集距離（またはレーベンシュタイン距離）は，二つの文字列の距離として，それらを一致させるために必要な文字の削除，挿入，置換操作の回数を距離として考える方法である．この距離はダイナミックプログラミングを用いて計算できる．次に，編集距離 DST_e の定義を示す．

定義 2.1　編集距離

α, β を文字列，x, y を異なる文字，“”を空文字列とする．関数 MIN は与えられた引数のうちもっとも小さい値を返す関数とする．

- 両方とも空文字のとき　　$DST_e(\text{“”}, \text{“”}) = 0.0$
- 長さ *1* 文字以下の異なる文字列のとき　　$DST_e(x, y) = 1.0$
- 先頭の *1* 文字が同じとき

$$DST_e(x\alpha, x\beta) = MIN(DST_e(\alpha, x\beta) + 1.0, DST_e(x\alpha, \beta) + 1.0, DST_e(\alpha, \beta))$$

- 先頭の *1* 文字が異なるとき *($x \neq y$)*

$$DST_e(x\alpha, y\beta) = MIN(DST_e(\alpha, y\beta) + 1.0, DST_e(x\alpha, \beta) + 1.0, DST_e(\alpha, \beta) + 1.0)$$

この定義で示すように，編集距離は不一致な文字を数えることによって二つの文字列の相違度を測っている．このため，編集距離は相違尺度とみることができる．定義から，編集距離は相違度が最小になるように，関数 MIN を用いて不一致な文字の位置を決定している．この部分が編集距離に表記の揺れに対応できる性質を持たせている．二つの文字列全体をみて，もっとも相違が小さくなるように試行錯誤することによって，表記の揺れがあっても類似したものとみなせる定義式となっている．

本論文では，情報検索に適した編集距離に基づく類似尺度を提案するための第一ステップとして，この定義を変形して，相違尺度である編集距離を類似尺度に変換する．具体的には，編集距離とは逆に類似度が最大になるように，関数 MIN の代わりに関数 MAX を用いて一致する文字の位置を決定する尺度に変換する．本論文では，この尺度を「単純編集類似度」と呼ぶことにした．この尺度と編集距離との違いは類似度を最大にするか相違度を最小にするかの違いだけなので，編集距離の持つ表記の揺れに寛容な性質は損なわれていない．次に単純編集類似度 SIM_1 の定義を示す．

定義 2.2 単純編集類似度

α, β を文字列，x, y を異なる文字，“”を空文字列とする．関数 MAX は与えられた引数のうちもっとも大きい値を返す関数とする．

- 両方とも空文字のとき　$SIM_1(“”, “”) = 0.0$
- 長さ *1* 文字以下の異なる文字列のとき　$SIM_1(x, y) = 0.0$
- 先頭の *1* 文字が同じとき

$$SIM_1(x\alpha, x\beta) = MAX(SIM_1(\alpha, x\beta), SIM_1(x\alpha, \beta), SIM_1(\alpha, \beta) + 1.0)$$

- 先頭の *1* 文字が異なるとき *($x \neq y$)*

$$SIM_1(x\alpha, y\beta) = MAX(SIM_1(\alpha, y\beta), SIM_1(x\alpha, \beta), SIM_1(\alpha, \beta))$$

2.2　文字重み編集類似度への拡張

単純編集類似度はすべての文字を同等に扱うため，一致する文字の重みはすべて 1.0 である．しかし，一致した文字が内容語に含まれる文字である場合と機能語に含まれる文字である場合とでは，情報検索においてその文字の貢献度は異なる．これは，意味に大きく関わる文字と表記の揺れとなりうる文字の貢献度の違いに相当する．一般に，情報検索の性能を向上させるために機能語を考慮せず，内容語だけを考慮するシステムが多く存在する．これは，機能語より

も内容語のほうが検索性能に貢献する度合いが高いためである．日本語における機能語はたとえば，「は」，「が」，「を」，「の」，「では」や「～する」，「～である」，「～した」，「～でない」，「～しない」などであり，主にひらがなで構成されている．一方，内容語は主に漢字やカタカナで構成されている．カタカナは主に外来語を構成している．このような背景から，情報検索に適した編集距離に基づく類似尺度を提案するための第二ステップとして，文字が一致した場合，常に 1.0 を加算するのではなく，一致した文字が持つ重みを加算する類似尺度に単純編集類似度を拡張する．言い換えると，この類似尺度は単純編集類似度の一般化である．本論文では，この類似尺度を「文字重み編集類似度」と呼ぶことにした．

また，編集距離（レーベンシュタイン距離）に関して各操作に重みを持たせた重みづきレーベンシュタイン距離がある (Kohonen 1995)．この距離では，操作ごとに対象となる文字に関する重みを持つ．ここで，文字に関して操作ごとに重み付けるのではなく，その文字に対してどの操作が行われても文字が持つ唯一の重みを付けると，編集距離（レーベンシュタイン距離）から単純編集類似度への変形のように，重みづきレーベンシュタイン距離を文字重み編集類似度に変形することができる．次に文字重み編集類似度 SIM_2 の定義を示す．

定義 2.3 文字重み編集類似度

α, β を文字列，x, y を異なる文字，“”を空文字列とする．関数 $Score(z)$ は文字 z が持つ重みを返す関数，関数 MAX は与えられた引数のうちもっとも大きい値を返す関数とする．

- 両方とも空文字のとき　　$SIM_2(“”, “”) = 0.0$
- 長さ *1* 文字以下の異なる文字列のとき　　$SIM_2(x, y) = 0.0$
- 先頭の *1* 文字が同じとき

$$SIM_2(x\alpha, x\beta) = MAX(SIM_2(\alpha, x\beta), SIM_2(x\alpha, \beta), SIM_2(\alpha, \beta) + Score(x))$$

- 先頭の *1* 文字が異なるとき $(x \neq y)$

$$SIM_2(x\alpha, y\beta) = MAX(SIM_2(\alpha, y\beta), SIM_2(x\alpha, \beta), SIM_2(\alpha, \beta))$$

例を用いて，単純編集類似度と文字重み編集類似度の振る舞いの違いを示す．

例 2.4 文字重みの効果

文 α, β, γ がそれぞれ

α：「機械で自動的に翻訳するシステム」

β：「自動翻訳システム」

γ：「人手で直感的に表示するシステム」

であり，*Score* 関数が次のように与えられたとする．

- 文字 x がひらがなの場合　　$Score(x) = 0.0$
- 文字 x がひらがな以外であった場合　　$Score(x) = 1.0$

表 1　文字重みの効果

引数	単純編集類似度	文字重み編集類似度
α, β	8.0	8.0
α, γ	9.0	5.0

このとき，α と β の類似度，α と γ の類似度は表 1のようになる．

この例において，α と β は人間であれば類似していると判断される文である．これらの文に対する類似度は一致する文字がすべてひらがな以外であるため，単純編集類似度でも文字重み編集類似度でも同じ類似度となる．一方，α と γ は人間であれば類似していないと判断される文である．これらの文に対して，単純編集類似度は一致する文字がひらがなであっても同じ重みを加算するため，高い類似度を与えてしまう．しかし，文字重み編集類似度は類似していると判断される α と β が持つ類似度よりも低い類似度を与えることができている．このことより，単純な重み付けでも，二つの文が類似するかしないかの判断に役立つことが予想できる．

2.3　文字列重み編集類似度への拡張

例 2.4の β は「自動翻訳システム」であり，この文は「自動」,「翻訳」,「システム」という三つの内容語で構成されている．これらの単語のうち，漢字で構成されるものは文字自体が意味を表しているために一文字だけでも検索に貢献する場合があるが,「システム」はほとんどの場合，一文字では意味を表すことができないカタカナで構成されている.「システム」は構成する文字が連続して現れることによって意味を表す単語となる．同様に，ひらがなで構成される文字列でも連続して現れることによって貢献する場合がある．このような場合，情報検索において，構成する文字それぞれの貢献よりも連続して現れることによって構成された文字列のほうが貢献度が高いことが知られている．言い換えると，一致した文字列を構成する文字毎に重みを加算するのではなく，文字列が持つ重みを加算したほうが検索性能が向上する可能性があるということである．多くの情報検索システムに用いられている単語を単位とし，一致した単語が持つ重みを加算することによって類似度を求める尺度があるが，この尺度もこの可能性に基づくものとみることができる．さらに，一単語より長い文字列が検索に大きく貢献するように *ngram* を単位とし，一致した *ngram* の重みを加算することによって類似度を求める尺度もある．この尺度は一致する文字列の長さに重きを置く尺度である．人間は一致する部分が長ければ長いほど，二つの文は類似していると判断することが多い．このため，一致する文字列の長さに重きを置くことは人間の直感に合致している．このような背景から，情報検索に適した編集距離に基づく類似尺度を提案するための最終ステップとして，一致した文字ではなく，一致した文字列が持つ重みを加算する類似尺度に文字重み編集類似度を拡張する．言い換えると，この類似尺度は文字重み編集類似度の一般化である．本論文では，この類似尺度を「文字列重

み編集類似度」と呼ぶことにした．次に文字列重み編集類似度 SIM_3 の定義を示す．

定義 2.5 文字列重み編集類似度

$\alpha, \beta, \gamma, \delta$ を文字列，ξ を長さ 1 以上の文字列，x, y を文字とする．関数 $Score(\epsilon)$ は文字列 ϵ が持つ重みを返す関数，関数 MAX は与えられた引数のうちもっとも大きい値を返す関数とする．

- 両方とも空文字のとき　　$SIM_3(\text{“”}, \text{“”}) = Score(\text{“”})$
- それ以外のとき

$$SIM_3(\alpha, \beta) = MAX(SIM_{3s}(\alpha, \beta), SIM_{3g}(\alpha, \beta))$$

  - 一致している最大の文字列を ξ として

$$SIM_{3s}(\xi\alpha, \xi\beta) = MAX(Score(\gamma) + SIM_3(\delta\alpha, \delta\beta))$$
$$for\ all\ \gamma;\ for\ all\ \delta; such\ that\ \xi = \gamma\delta$$

  - そのような文字列が存在しないとき　　$SIM_{3s}(\alpha, \beta) = 0.0$
  - 任意の文字列について

$$SIM_{3g}(x\alpha, y\beta) = MAX(SIM_3(\alpha, y\beta), SIM_3(x\alpha, \beta), SIM_3(\alpha, \beta))$$

3　本論文で用いる重み

例 2.4では，ひらがな以外の文字に重みを持たせたが，文字に持たせる重みを調節することによって，検索性能を大きく向上することが容易に予測できる．情報検索だけでなく他の分野においても，適した重みを決定することは難しく，多くの場合，経験によって決められることが多い．情報検索においては，検索対象となるデータによって調整することが広く行われている．本論文では，情報検索において重みの基本とされている文字列が一致したときの情報量に相当する IDF(Inverse Document Frequency) を用いることにした．また，通常の IDF は単語を対象とするが，文字列を対象とする IDF とした．これは，提案する類似尺度が $ngram$ を対象とするためである．本論文で扱う類似尺度が用いる重みはすべて IDF に基づくものとし，類似尺度の定義の違いによる検索性能の比較を行った．次に本論文で提案する類似尺度に用いた重みを返す関数 $Score$ を示す．ここで，$df(\xi)$ は長さ 1 以上の文字列 ξ が出現するドキュメントの数，N はドキュメントの総数とする．

- 空文字ならば，　　$Score(\text{“”}) = 0.0$
- それ以外ならば，　　$Score(\xi) = -\log_2(df(\xi)/N)$

本論文の目的は検索性能の低下を招く表記の揺れに寛容な類似尺度を提案することであるため，最適な重みについては今後の課題と考えている．

4　実験の概要

本論文では，提案する類似尺度の情報検索性能を評価するために，実験対象となる類似尺度を組み込んだ情報検索システムをそれぞれ作成し，検索性能を比較した．まず，単純編集類似度，文字重み編集類似度，文字列重み編集類似度の情報検索における性能を比較し，順に拡張したことによって表記の揺れに寛容な性質を損なうことなく，予想通り性能の向上を計ることができているかを検証する．そして，この検証においてもっとも高い性能を持つ類似尺度の検索性能を，多くの情報検索システムに用いられる類似尺度の基となっている二つの類似尺度と比較する．

実験では，情報検索テストコレクション NTCIR1(Kando et al. 1998; Kageura et al. 1997) を使用した．質問はキーワードではなく文章で表され，検索対象となる文書は技術論文の抄録である．コレクションには，質問集合 83 問（訓練用 30 問，本番用 53 問）と文書集合 33 万件，正解集合が含まれている．図 1と図 2に NTCIR1 の質問と文書の記述例をそれぞれ示す．質問

```
<検索課題 q=0020>
<タイトル>
カタカナ外来語
</タイトル>
<検索要求>
日本語文におけるカタカナ外来語の研究
</検索要求>
<検索要求説明>
カタカナ外来語の異表記の問題、原綴の類推、翻訳などについて論じたもの。英単語などの外国語のアルファベット表記からカタカナ表記への変換は除く、カタカナ文字の認識も除く。
</検索要求説明>
<概念>
カタカナ英語，カタカナ外来語，外来語，英文，かな，カタカナ，片仮名，カタカナ語
</概念>
<分野>
1.電子・情報・制御，8.人文・社会
</分野>
</検索課題>
```

図 1　質問の例

```
<REC>
<ACCN>gakkai-0000157215</ACCN>
<TITL TYPE="kanji">カタカナ表記の統一方式-予備分類とグラフ比較によるカタカナ表記のゆらぎ検出法-</TITL>
<AUPK TYPE="kanji">久保田 淳市 / 庄田 幸恵 / 河合 眞宏 / 玉川 博文 / 杉村領一</AUPK>
<CONF TYPE="kanji">研究発表会(自然言語処理)</CONF>
<CNFD>1993. 09. 16 - 1993. 09. 17</CNFD>
<ABST TYPE="kanji"><ABST.P>文章中のカタカナ表現を統一する効率的なカタカナ表記不統一検出方式を開発した。従来、辞書の見出しと文章中のカタカナ表記をゆらぎを許容して比較し、不統一箇所を検出する方式が多く提案されたが、カタカナ語は外来語を中心に新語が多く、予め辞書を整備しておくことは困難である。本方式では、基本的には辞書を使わず、カタカナ文字列同士のゆらぎ規則を文章中のカタカナ語に適用し、カタカナ表記のゆらぎを検出する。事前分類したカタカナ表記を有向グラフ形式の中間形式に変換し、効率良く比較検出を行う。試作と実験を行った結果、再現率97.4%適合率86.7%という結果を得た。</ABST.P></ABST>
<KYWD TYPE="kanji">カタカナ // 表記のゆれ // グラフ // 分類 // 推敲 // 再現率</KYWD>
<SOCN TYPE="kanji">情報処理学会</SOCN>
</REC>
```

図 2　文書の例

の記述は「タイトル」,「検索要求」,「検索要求説明」,「概念」,「分野」で構成され，検索対象となる文書の記述は「タイトル」,「著者名」,「抄録」,「分野」などで構成されている．本論文では，質問の「検索要求」部分の文章と，文書の「抄録」部分を連結した文章との類似度を測ることによって，情報検索を行った．

システムから得る結果は，質問ごとに類似度を高い順に並べたランキングリストである．性能評価は，この実験結果の上位1000件に対する11点平均精度を比較することによって行った．

5　編集類似度の性質および検索性能の検証

本節では，単純編集類似度 (SIM_1)，文字重み編集類似度 (SIM_2)，文字列重み編集類似度 (SIM_3) の情報検索における性能を比較し，順に拡張したことによって表記の揺れに寛容な性質を損なうことなく，予想通り性能の向上を計ることができているかを検証する．実験結果を表2–表4に示す．まず，情報検索性能について検証する．表2は，単純編集類似度，文字重み編集類似度，文字列重み編集類似度をそれぞれ用いたシステムで訓練用30問について情報検索を行った結果の11点平均精度 (11 point average precision) に示す．11点平均精度は情

表 2　編集類似度の11点平均精度（訓練課題30問）

類似尺度	SIM_1	SIM_2	SIM_3
11-pt	0.135	0.203	0.281

表 3　質問ごとの11点平均精度

質問番号	SIM_1	SIM_2	SIM_3	質問番号	SIM_1	SIM_2	SIM_3
1	0.2611	0.2675	0.3006	16	0.5577	0.6777	0.8263
2	0.0159	0.6025	0.6704	17	0.0009	0.0283	0.0407
3	0.0006	0.0023	0.0148	18	0.0380	0.1288	0.0980
4	0.0049	0.2929	0.3136	19	0.4026	0.4487	0.6890
5	0.0006	0.0009	0.0013	20	0.2683	0.6493	0.7775
6	0.0212	0.1626	0.1890	21	0.0215	0.1339	0.1583
7	0.0022	0.0000	0.0024	22	0.2068	0.2525	0.2493
8	0.0020	0.0003	0.2119	23	0.2308	0.3596	0.3481
9	0.1050	0.0398	0.2402	24	0.2903	0.3153	0.3234
10	0.1052	0.3669	0.3618	25	0.0248	0.1324	0.4877
11	0.3268	0.2194	0.2638	26	0.4438	0.1266	0.5136
12	0.0742	0.0224	0.1566	27	0.0259	0.1784	0.2556
13	0.0057	0.0330	0.0655	28	0.0478	0.0518	0.0589
14	0.2980	0.3646	0.4257	29	0.1753	0.0501	0.1151
15	0.1001	0.1528	0.2116	30	0.1025	0.0349	0.0527

表 4　類似尺度ごとの比較

X vs Y	X win	Y win
SIM_1 vs SIM_2	8	22
SIM_2 vs SIM_3	4	26
SIM_1 vs SIM_3	3	27

報検索における一般的な評価基準で，再現率 (recall) が 0%，10%，20%，30%，40%，50%，60%，70%，80%，90%，100%の 11 点における適合率 (precision) の平均値である (Manning, Schutze 1999)．この表はこの実験における各システムの検索性能を表し，文字列重み編集類似度が単純編集類似度や文字重み編集類似度より情報検索に有効であることを示している．表 4 は，訓練用 30 問において一方の編集類似度が他方の編集類似度より検索に有効であった質問を表 3を用いて数えた結果を示す．この表は，この実験において，文字重み編集類似度が単純編集類似度よりも多くの質問に対して有効に働き，文字列重み編集類似度が単純編集類似度や文字重み編集類似度よりもさらに多くの質問に対して有効に働くことを示している．たとえば，「各質問について SIM_2 より SIM_1 が高い 11 点平均精度を出す確率が 1/2 以下のとき，SIM_2 が SIM_1 より 30 個のうち 22 個以上の割合で性能が高い」という仮説を立てた場合，危険率 8.1×10^{-3} 以下で仮説は棄却される．これは，SIM_2 が SIM_1 より高い性能を出す確率は 1/2 以上であることを示し，SIM_2 と SIM_1 には有意な差があることがわかる．また同様に，SIM_3 と SIM_1 には危険率 3.0×10^{-5} 以下のレベルで有意な差があり，SIM_3 と SIM_2 には危険率 4.2×10^{-6} 以下のレベルで有意な差があることがわかる．以上のことから，編集距離に基づく類似尺度を拡張することによって情報検索性能を向上していることが確認できる．

次に，表記の揺れに寛容な性質が損なわれていないかを検証する．例として，図 1に示す質問 20 を取り上げる．検索に使うこの質問の検索要求は「日本語文におけるカタカナ外来語の研究」である．図 2に示す文書は質問 20 に関連する文書である．まず，単純編集類似度から文字重み編集類似度に拡張した場合，表記の揺れに寛容な性質が損なわれていないかを検証する．これらの質問と文書において単純編集類似度と文字重み編集類似度は同じ「文，る，カ，タ，カ，ナ，外，来，語，の」の 10 文字が一致する．重みが影響するのは，質問に現れる文字が文書ではそれらの文字が質問に現れる順とは異なり，前後に交換されている場合に考えられる．たとえば，質問に「有無」が現れ，文書に「‥は無いが，‥は有る」と,「有」と「無」が逆順に現れるとき，もし「無」が持つ重みのほうが大きい場合，文字重み編集類似度は前にある「有」ではなく，後ろにある「無」の重みを加算するが，単純編集類似度はどちらを選んでも同じである．このため，単純編集類似度と文字重み編集類似度において，稀に選択された文字が異なると考えられるが，サンプルで調査したすべての質問においてはすべて同じ文字を選択していた．これは，単純編集類似度を文字重み編集類似度に拡張しても表記の揺れに寛容な性質を保持しているということを表している．図 2の文書を，単純編集類似度は 181 位，文字重み編集類似度は 10 位に位置付けている．表 3に示す質問 20 における 11 点平均精度を見ると，単純編集類似度よりも文字重み編集類似度のほうが精度が高い．これは，表記の揺れに寛容な性質を持ち，一致した文字の重みを考慮したことによって検索性能が向上したことを示している．

一方，文字列重み編集類似度は「文，る，カタ，カナ，外，来語，の」の 7 つの文字列が一致する．これは，単純編集類似度と文字重み編集類似度で選択された 10 文字と同じである．これ

は，文字重み編集類似度を文字列重み編集類似度に拡張しても表記の揺れに寛容な性質を保持しているということを表している．実際に，図2の文書に対して，文字重み編集類似度は4.44，文字列重み編集類似度は12.37を得ている．また，この文書を文字重み編集類似度は10位に位置付けているが，文字列重み編集類似度は6位に位置付けている．表3に示す質問20における11点平均精度を見ると，文字重み編集類似度よりも文字列重み編集類似度のほうが精度が高い．これは，表記の揺れに寛容な性質を持ち，一致した文字列の重みを考慮したことによって検索性能が向上したことを示している．

以上のように，類似尺度を拡張しても表記の揺れに寛容な性質を損なっていないことをサンプルで確認した．

6　基本的な類似尺度との検索性能比較

5節に示した実験において，本論文で提案する三つの編集類似度のなかで，文字列重み編集類似度がもっとも情報検索において有効であることがわかった．本節では，提案する文字列重み編集類似度の検索性能を，多くの情報検索システムに用いられる類似尺度の基となっている類似尺度と比較する．まず，情報検索において基とされる類似尺度を用いたシステムの概要を示す．本論文では，形態素解析を利用して内容語を抽出し，質問と文書のどちらともに存在する内容語の重みを加算する類似尺度と，質問と文書のどちらともに存在する文字列 (*ngram*) の重みを加算する類似尺度を選び，これらの尺度を用いた二つのシステムをベースラインシステムとした．

6.1　ベースラインシステム：BD

本論文では，一つ目のベースラインシステムとして，形態素解析を利用し内容語を抽出し，質問と文書のどちらともに存在する内容語の重みを加算することによって類似度を測るシステムを作成した．本論文では，このシステムを辞書を用いることから「BD(baseline-dictinary)」と呼ぶことにした．形態素解析には，日本語形態素解析プログラム「茶筌」(Matsumoto, Kitauchi, Yamashita, Hirano 1997) を用いた.「茶筌」は大きな日本語の単語辞書を使って文字の列を単語に区切り，品詞を割り当てるシステムである．

BD システムはまず，質問と文書それぞれに対して形態素解析を行い，そして，解析結果から内容語（名詞，動詞，未定義語）の原形を抽出し，類似度を測るための内容語だけが並ぶ質問集合のファイルと文書集合のファイルを作成する．BD システムに原形を採用した理由は，文章に現れる内容語の形そのままを対象とした場合と原形を対象とした場合の検索性能を比較した結果，原形のほうが検索性能が高かったためである．これらの二つのファイルを用いて，Salton(Salton, Buckley 1988) が用いている多くの情報検索手法の基となっている内積スコア

リング関数を用いて類似度を求める．内積スコアリング関数は文字列重み編集類似度に用いられる重み関数と同じ $IDF(=-\log(df/N))$ に基づく重み関数である．このことから，辞書を用いて形態素解析を行うことに関して比較することにした．次に BD システムに用いた類似尺度 SIM_{dict} の定義を示す．

定義 6.1 t は比較される各々の文字列両方に現われる単語，$tf(t)$ はそのドキュメントの単語 t の出現頻度 *(term frequency)*，$df(t)$ は単語 t が出現するドキュメントの数 *(document frequency)*，N はドキュメントの総数とする．

$$SIM_{dict} = \sum_{t} tf(t) \cdot (-log_2(df(t)/N))$$

6.2　ベースラインシステム：BN

本論文では，二つ目のベースラインシステムとして，質問と文書のどちらともに存在する文字列 (*ngram*) の重みを加算することによって類似度を測るシステムを作成した．本論文では，このシステムを *ngram* をマッチングの対象とすることから「BN(baseline-*ngram*)」と呼ぶことにした．BN システムは，質問と文書のどちらともに現れる文字列を検出し，文字列重み編集類似度に用いられる Score 関数を用いて類似度を求める．通常，処理効率を考え，長さ 2 のバイグラム (bigram) または長さ 3 のトライグラム (trigram) のような短い *ngram* だけをマッチングの対象とするが，本論文では，SIM_3 がすべての *ngram* を対象としているので，条件をそろえるために，すべての *ngram* を対象とすることにした．一般には，(Fujii, Croft 1993) が示したように，短い *ngram* は日本語にはかなり効果的であることが報告されている．しかし，実際に NTCIR1 において短い *ngram* に制限した場合と制限しない場合の検索性能の比較を行った結果，制限しない場合のほうが高い性能を得たため，バイグラムやトライグラムより長い *ngram* を考慮することにした．また，提案する文字列重み編集類似度においても扱う文字列の長さを制限していないので，条件は同じである．

実際に長い *ngram* を考慮することは，複合語のマッチングを行う情報検索 (山田，森，中川 1997) の報告で，共起情報を用いないケースに相当する．DP も長い *ngram* を検出するので，BN を比較対象とした．BN システムはマッチングの対象が文字列重み編集類似度と同じ文章にある部分文字列 (*ngram*) である．BN システムと文字列重み編集類似度の唯一の違いは，類似尺度の定義が語順を無視した重みの総和をとるか語順を保存した重みの合計の最大値をとるかの違いである．このことが表記の揺れに寛容な性質を持つか持たないかの差となっていると予想される．このため，本論文では，この二つの類似尺度を表記の揺れに対する振る舞いについて比較することにした．次に BN に用いた類似尺度 SIM_{ngram} の定義を示す．

定義 6.2 α, β, ξ, η を文字列とする．α_{ik} を i 番目の文字から $i+k-1$ 番目の文字までの α の部分文字列とし，β_{jk} を j 番目の文字から $j+k-1$ 番目の文字までの β の部分文字列とする．

また，$Score$ は IDF に基づく文字列から正の実数値を求める関数とする．

$$SIM_{ngram} = \sum_{i,j,k} Comp(\alpha_{ik}, \beta_{jk})$$

ただし，$Comp(\xi, \eta)$ は次のように定義され，ここで現れる $Score(\xi)$ は *2.3*節に示したものと同じである．

- $\xi = \eta$ ならば，　$Score(\xi) = -log_2(df(\xi)/N)$
- $\xi \neq \eta$ ならば，　0.0

6.3 検索性能の比較

本節では，本論文で提案する文字列重み編集類似度 (SIM_3) を用いたシステムと，SIM_{dict} を用いた BD システム，SIM_{ngram} を用いた BN システムの検索性能を比較する．本論文では，文字列重み編集類似度を用いたシステムをダイナミックプログラミングを用いて計算できることから，「DP」と呼ぶことにした．

表 5　システムの 11 点平均精度（訓練課題 30 問）

システム	BD	BN	DP
11-pt	0.154	0.164	0.281

表 6　それぞれのシステムの質問ごとの 11 点平均精度

質問番号	BD	BN	DP	質問番号	BD	BN	DP
1	0.2462	0.2216	0.3006	16	0.0180	0.3275	0.8263
2	0.2099	0.5022	0.6704	17	0.0471	0.0011	0.0407
3	0.0257	0.0028	0.0148	18	0.0788	0.0051	0.0980
4	0.1048	0.1815	0.3136	19	0.2091	0.4901	0.6890
5	0.1046	0.0008	0.0013	20	0.5660	0.5705	0.7775
6	0.0580	0.0221	0.1890	21	0.0881	0.0713	0.1583
7	0.0005	0.0010	0.0024	22	0.0516	0.0703	0.2493
8	0.0836	0.0086	0.2119	23	0.2003	0.1475	0.3481
9	0.0157	0.0372	0.2402	24	0.1915	0.2887	0.3234
10	0.3751	0.2991	0.3618	25	0.6076	0.2291	0.4877
11	0.2533	0.0303	0.2638	26	0.1087	0.4665	0.5136
12	0.1008	0.1687	0.1566	27	0.0659	0.0696	0.2556
13	0.0707	0.0150	0.0655	28	0.0072	0.0435	0.0589
14	0.4107	0.3051	0.4257	29	0.1543	0.0882	0.1151
15	0.1538	0.1436	0.2116	30	0.0094	0.0142	0.0527

表 7　システムごとの比較

X vs Y	X win	Y win
DP vs BD	23	7
DP vs BN	29	1
BD vs BN	15	15

表5に33万件のドキュメントに対して30個の検索質問を行った場合の11点平均精度を示す．この表は実験において，DPがBDやBNよりも精度が高かったことを示している．表7はそれぞれのシステムを二つずつ各質問について，表6に示す11点平均精度を使って比較し，すべての質問について数値で判定した結果から作成した．これらの表もまた，DPがBDやBNよりも精度が高いことを示している．

質問1に対して，三つのシステムは同じような性能を示す．質問1は，用語と用語を構成する単語の多くがそれらのIDF重みによって示されるようなよいキーワードである用語を含む質問であり，すべてのシステムにとって簡単な質問である．図3の左図に上位1000件の文書に関する再現率を示す．この図から，三つのシステムは比較の条件が揃っていることがわかる．実験結果において，表7から5節に示すような仮説を立て，検索性能を検証すると，DPとBDには危険率2.6×10^{-3}以下のレベルで有意な差があり，DPとBNには危険率2.9×10^{-8}以下のレベルで有意な差があることがわかる．

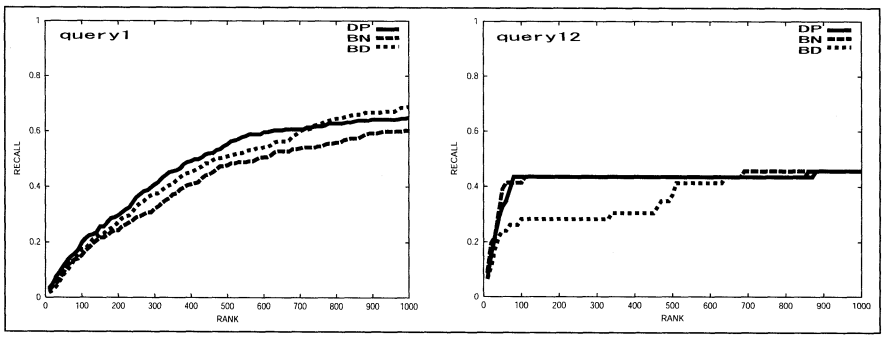

図 3　再現率：質問1「自立移動ロボット」，質問12「データマイニング」

6.4　BDとの振る舞いの差

本論文の実験において，表7に示すように，DPはBDと同等以上の性能を持つことを示している．これは，未知語に対する振る舞いの違いによるものである．辞書を利用するシステムは解析できない未知語が重要となる質問に対応することが難しい．たとえば，図3の右図はそのような質問に関する再現率のグラフである．BDはデータマイニングを「デー」「タマ」「イニング」に分割してしまうため，その結果，情報検索の性能が低い．NTCIR1において，このように未知語が重要となる質問がこの他にも存在する．辞書を利用するBDの場合，新しい単語が作成されるたびに辞書に単語を登録すれば語分割の失敗を避けることができるが，新しい単語に対する辞書のメンテナンスが必要である．一方，BNとDPは辞書を利用しないため，このコストがかからないという利点を持っている．しかし，未知語を学習することによって検索性能が向上することは明白であるため，システムを相補的に用いることが理想である．

6.5 実行時間

実験における各システムの実行時間を表8に示す．BDはperlで記述し，他のシステムはCで記述したため，BDシステムの実行時間は参考値であり，前処理となる文書頻度(document frequency)の計算を除いた類似度計算のみの実行時間である．また，すべてのシステムは文書

表 8　実行時間

システム	**BD**	**BN**	SIM_1	SIM_2	**DP**(SIM_3)
総実行時間 [sec]	43947	2099	3257	3713	5302
1質問当たり [sec]	1464.9	70.0	108.6	123.8	176.7
1ドキュメント当たり [msec]	4.44	0.21	0.33	0.38	0.54

と質問を一つずつ比較するシステムであるため，インデックスファイルは使用していない．実験はVine Linux 2.0，CPU800MHz，メモリ1GBの計算機を用いて行った．この表では，“総実行時間”は33万件のドキュメントに対して30個の質問を検索することにかかった時間，“1質問当たり”は総実行時間を30で割った，1質問当たりにかかる平均実行時間，“1ドキュメント当たり”は1質問当たりの時間を33万で割った，1ドキュメントとの類似度を計算することにかかる平均実行時間である．BNと$DP(SIM_3)$を比較すると，実際のデータでの計算時間の差は2倍程度であった．これは，任意の文字列の文書頻度を高速に求められる方法(Yamamoto, Church 1998)を利用した効果である．そして，文字列重みを利用しても単純な編集距離の計算よりも桁違いに遅くないことがわかる．しかし，提案する類似尺度を用いたシステムは情報検索システムとして実用的とは言えない．本論文では，情報検索に利用できる表記の揺れに寛容な類似尺度の提案を目的としているので，インデックスを利用する処理速度の向上は今後の課題と考えている．

7 考察

7.1 単語の順序について

定義2.5に示すように，文字列重み編集類似度は一致する文字列の重みにその文字列以外の部分の類似度を加算する操作をしている．この定義では，質問と文書に二つずつ同一の文字が出現した場合に，出現の順序が一致したときには加算操作の対象になるが，順序が異なった場合には，加算の対象とならないため，結果的にどちらかの重みが選ばれることになる．この「順序が一致したときに，スコアが加算される」という性質は単純編集類似度，文字重み編集類似度，文字列重み編集類似度に共通する性質である．順序関係を保存しながら，重みの合計が最大になるように部分文字列の組合せを探す尺度のなかで文字列重み編集類似度はすべての文字列の重みを考慮するということが特徴となっている．キーワードの検索においても順序を保存することで検索精度が向上するという報告(田中 1997)があるが，文字列重み編集類似度はキー

ワードに限定せず質問を構成する文字列の順序を保存する．

また，順序情報を利用するという意味では，形態素解析を行い，内容語の列を作り，その列が持つ順序情報を利用して検索性能を向上させている研究 (大竹，増山，山本 1999) があるが，本論文で提案する手法では形態素解析を行わない．さらに，修飾語の欠損と付加がある場合にも提案する手法は対応できる．

7.2　句の検出

文字列重み編集類似度は，重みの合算の過程において類似判定に効果のある部分文字列を選び出す処理を行っている．つまり，定義2.5に示される定義式で関数 MAX によって選び出された文字列は類似判定に効果がある文字列として選び出されている．この一連の文字列は，検索に効果があるひとかたまりと解釈できる．言い替えれば，文字列重み編集類似度によって，検索に利用可能な「分離している複合語」を抽出していると解釈できる．これは，類似判定ごとに「語」の定義を変更することで効果を上げている情報検索システム (小澤，山本，山本，梅村 1999) と同様に，情報検索のための句分割方法の一つと解釈することもできる．

検索に効果がある文字列の集合を選ぶためによく用いられる方法は共起関係を利用する方法である (高木，木谷 1996)．文字列重み編集類似度で選び出される文字列の集合は共起によるものとは異なる文字列となる．端的には，文字列重みによるものは，IDF が高ければ統計的に独立に出現し，共起関係にない文字列でもキーワードとして検出される．実際に，文字列重み編集類似度で選択された一群の文字列の性質を分析することは行う価値のある今後の課題である．

7.3　シソーラスの利用

文字列の一致の検出を行うときに，シソーラスを利用する拡張が考えられる．すなわち，質問中の文字列がシソーラスの見出し語にあるときは，その対応する単語がドキュメントにあったときに文字列が一致しているとみなす拡張である．この方法を実装することは簡単であるが，適切な重みを何にするか，実際に検索に効果があるかなどの問題が存在し，今後の課題である．

8　まとめ

本論文では，情報検索のための表記の揺れに寛容な類似尺度を提案し，その検索性能を評価した．表記の揺れに対応できる尺度として編集距離が知られているが，実際にこの尺度を単純に類似尺度に変換したものでは性能がでない．本論文では，表記の揺れに寛容な類似尺度を，情報検索に適するように，文字列に対する重みを計算するように拡張した．それを用いて情報検索による評価を行い，性能が向上したことを示した．さらに，提案する類似尺度を組み込んだ情報検索システムを構築し，多くのシステムに用いられている一般的な類似尺度と同等以上

の検索性能を実現できたことを示した．

謝辞

本研究は住友電気工業株式会社の援助による成果です．そして，本研究を進めるにあたって有意義なコメントを戴いたAT&TのKenneth W. Church氏と筑波大学の山本幹雄助教授に深く感謝いたします．

参考文献

Hideo Fujii and W. Bruce Croft (1993). "A Comparison of Indexing Techniques for Japanese Text Retrieval ." In *Proceeding of SIGIR'93*, pp. 237–246, Pittsburgh PA, USA.

Kyo Kageura, Teruo Koyama, Masaharu Yoshioka, Atsuhiro Takasu, Toshihiko Nozue, and Keita Tsuji (1997). "NACSIS Corpus Project for IR and Terminological Research." In *Natural Language Proceeding Pacific Rim Symposium'97*, pp. 493–496.

Noriko Kando, Teruo Koyama, Keizo Oyama, Kyo Kageura, Masaharu Yoshioka, Toshihiko Nozue, Atsushi Matsumura, and Kazuko Kuriyama (1998). "NTCIR:NACSIS Test Collection Project." In *20th Annual Colloquium of BCS-IRSG.*

Teuvo Kohonen (1995). "Self-Organizing Maps." Springer-Verlag Berlin Heidelberg.

Robert R. Korfhage (1997). "Information Storage and Retrieval ." In *WILEY COMPUTER PUBLISHING*, pp. 291–303, John Wiley & Sons, Inc.,Printed in USA.

Christopher D. Manning and Hinrich Schutze (1999). "Foundationals of Statistical Natural Language Processing" The MIT Press, Cambridge MA.

Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Osamu Imaichi, and Tomoaki Imamura (1997). "Japanese Morphological analysis System ChaSen Manual." *NAIST Technical Report, NAIST-IS-TR97007*, http://cactus.aist-nara.ac.jp/lab/nlt/chasen.html.

大竹清敬，増山繁，山本和英 (1999). "名詞の連接情報を用いた関連文書検索手法." 情報処理学会論文誌, **40(5)**, pp. 2460–2467.

小澤智裕，山本幹雄，山本英子，梅村恭司 (1999). "情報検索の類似尺度を用いた検索要求文の単語分割." 言語処理学会大会, **A5-2**.

Gerard Salton and Christopher Buckley (1988). "Term-Weighting Approaches in Automatic Text Retrieval ." In *Information Proceeding and Management*, **24**, pp. 513–523.

高木徹，木谷強 (1996). "単語出現共起関係を用いた文書重要度付与の検討." 情報処理学会 情報学基礎研究会, **FI41-8**.

田中英輝 (1997). "長い日本語表現の高速類似検索手法." 情報処理学会 自然言語処理研究会, **NL121-10**.

山田剛一，森 辰則，中川 裕志 (1997). “複合語マッチングと共起情報を併用する情報検索.” 情報処理学会論文誌, **39(8)**, pp. 2431–2439.

Mikio Yamamoto and Kennth W. Church (1998). “Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus.” In *Proceeding of the Sixth Workshop on Very Large Corpora*, Editor: Eugene Charniak, pp. 28–37.

略歴

山本 英子: 1996 年豊橋技術科学大学情報工学課程卒業．2002 年同大学大学院工学研究科電子・情報工学専攻博士課程修了．博士（工学）．同年，通信総合研究所に専攻研究員として入所．

武田 善行: 2000 年豊橋技術科学大学工学部情報工学課程卒業．2002 年同大学工学研究科情報工学専攻修士課程修了．同年同大学大学院工学研究科電子・情報工学専攻博士課程入学．現在に至る．

梅村 恭司: 1983 年東京大学大学院工学系研究系情報工学専攻修士課程修了．博士（工学）．同年，日本電信電話公社電気通信研究所入所．1995 年豊橋技術科学大学工学部情報工学系助教授，現在に至る．システムプログラム，記号処理の研究に従事．ACM，ソフトウェア科学会，電子情報通信学会，計量国語学会各会員．

(2002 年 5 月 30 日 受付)
(2002 年 9 月 11 日 再受付)
(2002 年 10 月 4 日 採録)